

**Amendments to the Drawings:**

The drawings were objected to because of inconsistencies in FIG. 1 and the specification. Further FIG. 3A and FIG. 3B were found to contain misspelled words. Accordingly, FIG. 1 has been amended to properly label mislabeled features of the invented system and FIG. 3A and FIG. 3B has been amended to correct the misspelled words. Reconsideration and allowance of the replacement sheets of drawings is respectfully requested.

Attachment:            Replacement Sheets

## REMARKS

### **Allowed claims:**

Applicants acknowledge and appreciate allowance of claim 24.

### **Status of Claims:**

Claims 1-28 were rejected. Claims 1, 3, 5, 7-10, 12-16, 18, 19, and 22-28 have been amended. Claim 2 was canceled. Reconsideration and allowance of claims 1, 3, 5, 7-10, 12-16, 18, 19, and 22-28 is respectfully requested.

### **Claim Rejections:**

Claims 1-6 and 15 were rejected under 35 USC 112, as being indefinite. Herein, claims 1, 3, 5, and 15 have been amended and are believed to moot the section 112 rejections. Claim 2 was canceled. No new matter was added. Accordingly, reconsideration and allowance of claims 1, 3, 5, and 15 is respectfully requested.

Claims 1-13 were rejected under 35 USC 103 as being unpatentable over Noland et al., U.S. Patent No. 7,080,378, hereinafter *Noland*; in view of Kawamoto et al., U.S. Patent No. 7,117,499, hereinafter *Kawamoto*; in view of Sexton et al., U.S. Patent No. 6,854,114, hereinafter *Sexton*; and further in view of Gootherts U.S. Patent Publication No. US 2002/0099759 A1, hereinafter *Gootherts*.

*Noland* discloses a method for scaling resources according to workload among virtual servers running on a mainframe computer. The method includes monitoring the number of service requests received by a cluster of virtual servers and determining if the service requests exceed a specified service limit for the servers. If the number of service requests exceeds the specified service limit, a new virtual server is automatically deployed by a software solution. This additional virtual server performs identical services as the other virtual servers in the cluster. This process is repeated until there are a sufficient number of servers to handle the workload. Service requests are then allocated

among the cluster of virtual servers, until the number of requests falls below a certain threshold, at which point the extra servers are automatically deactivated.

*Kawamoto* discloses a virtual computer system and method where computer resources are automatically and optimally allocated to logical partitions according to loads to be accomplished by operating systems in the logical partitions and setting information based on a knowledge of workloads that run on the operating systems. Load measuring modules are installed on the operating systems in order to measure the loads to be accomplished by the operating systems. A manager designates the knowledge concerning the workloads on the operating systems through a user interface. An adaptive control module determines the allocation ratios of the computer resources relative to the logical partitions according to the loads and the settings, and issues an allocation varying instruction to a hypervisor so as to thus instruct variation of allocations.

*Sexton* discloses techniques for instantiating separate Java virtual machines for each session established by a server. Since each session has its own virtual machine, the Java programs executed by the server for each user connected to the server are insulated from the Java programs executed by the server for all other users connected to the server. The separate VM instances can be created and run, in separate units of execution that are managed by the operating system of the platform on which the server is executing. The separate VM instances may be executed either as separate processes, or using separate system threads. Because the units of execution used to run the separate VM instances are provided by the operating system, the operating system is able to ensure that the appropriate degree of insulation exists between the VM instances.

*Gootherts* discloses a method and apparatus for balancing processing loads to avoid starvation of threads. The disclosed method of load balancing evaluates the load and state of multiple processors. If at least one processor is in a source state and at least one processor is in a sink state, the processing load is balanced to avoid starvation. A thread is transferred from the heaviest loaded, source state processor to the least loaded, sink state processor. Each processor load and state is then reevaluated and, if needed, the load balancing with starvation avoidance repeated.

Thus, the method disclosed by *Noland* includes monitoring service requests and if the number of service requests exceeds the specified limit, adding virtual servers to a cluster until the need is met, and when the number of requests falls below a certain threshold, at which point the extra servers are automatically deactivated. *Kawamoto* discloses automatically and optimally allocating computer resources to logical partitions according to loads to be accomplished by operating systems in the logical partitions and setting information based on a knowledge of workloads that run on the operating systems. *Sexton* discloses techniques for instantiating separate Java virtual machines for each session established by a server. Separate VM instances can be created and run, in separate units of execution that are managed by the operating system of the platform on which the server is executing. The separate VM instances may be executed either as separate processes, or using separate system threads.

Claim 1 was amended to recite:

A system to provide finer grain control in optimizing multiple workloads across multiple servers, comprising:  
a plurality of servers to be utilized by multiple workloads;  
a plurality of virtual machines at each of the plurality of servers, wherein the plurality of virtual machines at each of the plurality of servers each serve a different one of the multiple workloads; and  
resource management logic to distribute server resources to each of the plurality of virtual machines according to current and predicted resource needs of each of the multiple workloads utilizing the server resources,  
whereby, each of the multiple workloads are distributed across the plurality of servers, wherein fractions of each of the multiple workloads are handled by the plurality of VMs virtual machines,  
whereby, the fractions of each of the multiple workloads handled by each of the VMs virtual machines can be dynamically adjusted to provide for optimization of the server resources utilized by the multiple workloads across the multiple servers.

As claimed in amended claim 1, applicant's system includes "... a plurality of virtual machines at each of the plurality of servers, wherein the plurality of virtual machines at each of the plurality of servers each serve a different one of the multiple workloads..." Nowhere in the cited prior art, taken alone or in any combination, is a system that includes plurality of virtual machines at each of the plurality of servers each

serve a different one of the multiple workloads. Thus, applicant's invented system is not obvious in view of the cited prior art references, taken alone or in combination, and is patentable thereover. Reconsideration and allowance of claim 1 is respectfully requested.

Claims 2-13 depend from amended independent claim 1. Claim 2 has been canceled. Amended independent claim 1 is now believed to be patentable over the cited prior art references and Claims 3-13 further define the patentable subject matter of their base claim, amended independent claim 1. Reconsideration and allowance of claims 3-13 is requested.

Claims 14, 16-18, 20-23, 25, and 27 were rejected under 35 USC 103(a) as being unpatentable over *Noland* in view of *Kawamoto*. Claims 14, 16, 18, 20-23, 25, and 27 have been amended.

As noted above, the method disclosed by *Noland* includes monitoring service requests and if the number of service requests exceeds the specified limit, adding virtual servers to a cluster until the need is met, and when the number of requests falls below a certain threshold, at which point the extra servers are automatically deactivated. *Kawamoto* discloses automatically and optimally allocating computer resources to logical partitions according to loads to be accomplished by operating systems in the logical partitions and setting information based on a knowledge of workloads that run on the operating systems.

Claim 14 was amended to recite:

A server optimization device, for providing finer grain control in a virtual machine based hosting architecture, comprising:  
at least one load balancer component to identify resource requirements for multiple different workloads in the VM based hosting architecture;  
a global resource allocator partitioning component to assign virtual machines from multiple server machines to a workload according to the identified resource requirements; and  
the global resource allocator partitioning component to assign resources at each of the multiple server machines to the assigned virtual machines according to the identified resource requirements.

Claim 25 was amended similarly. As recited in amended claim 14, applicant's optimization device includes "...a global resource allocator partitioning component to assign virtual machines from multiple server machines to a workload according to the identified resource requirements..." The global resource allocator includes a "...partitioning component to assign resources at each of the multiple server machines to the assigned virtual machines according to the identified resource requirements." Accordingly an optimization device that includes a global resource allocator includes a "...partitioning component to assign resources at each of the multiple server machines to the assigned virtual machines according to the identified resource requirements is not obvious in view of the teachings of the cited prior art.

Claims 16-18 and 20-23 depend from amended independent claim 14 and claim 27 depends from amended independent claim 25. Amended independent claims 14 and 25 are now believed to be patentable over the cited prior art references. Claims 16-18 and 20-23 further define the patentable subject matter of their base claim, amended independent claim 14 and claim 27 further defines the patentable subject matter of its base claim, amended independent claim 25. Reconsideration and allowance of claims 16-18, 20-23, and 27 is respectfully requested.

Regarding Claim 28, claim 28 was amended to claim:

A computer program product for use with a computer hosting architecture, for providing finer grain control in a virtual machine based hosting architecture, comprising:  
    a computer-readable medium  
    means, provided on the computer-readable medium, for identifying resource requirements for multiple different workloads in the virtual machine based hosting architecture;  
    means, provided on the computer-readable medium, for assigning virtual machines from multiple server machines to a workload according to the identified resource requirements; and  
    means, provided on the computer-readable medium, for assigning resources at each of the multiple server machines to the assigned virtual machines according to the identified resource requirements.

*Noland* discloses a computer program product that includes monitoring service requests and if the number of service requests exceeds the specified limit, adding virtual

servers to a cluster until the need is met, and when the number of requests falls below a certain threshold, at which point the extra servers are automatically deactivated.

*Kawamoto* discloses automatically and optimally allocating computer resources to logical partitions according to loads to be accomplished by operating systems in the logical partitions and setting information based on a knowledge of workloads that run on the operating systems. *Gootherts* discloses a method for balancing processing loads to avoid starvation of threads. The disclosed method of load balancing evaluates the load and state of multiple processors and if at least one processor is in a source state and at least one processor is in a sink state, the processing load is balanced to avoid starvation.

However, amended claim 28 recites “means, provided on the computer-readable medium, for identifying resource requirements for multiple different workloads in the virtual machine based hosting architecture; means, provided on the computer-readable medium, for assigning virtual machines from multiple server machines to a workload according to the identified resource requirements; and means, provided on the computer-readable medium, for assigning resources at each of the multiple server machines to the assigned virtual machines according to the identified resource requirements.” Thus, a computer program product that includes means for assigning resources at each of the multiple server machines to the assigned virtual machines according to the identified resource requirements is not obvious in view of the cited prior art references taken alone or in combination. Reconsideration and allowance of claim 28 is respectfully requested.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 927-3387. For payment of any additional fees due in connection with the filing of this paper, the Commissioner is authorized to charge such fees to Deposit Account No. 09-0441 (Order No. ARC920030046US1).

Respectfully submitted,

By: /Jeffrey P. Aiello/

Jeffrey P. Aiello  
Reg. No. 39,086  
Intellectual Property Law  
Systems & Technology Group  
IBM  
650 Harry Road  
San Jose, California 95120  
Telephone: (408) 927-3387  
Facsimile: (408) 927-3375